Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 1997 Proceedings

Americas Conference on Information Systems (AMCIS)

8-15-1997

Toward a Framework for Understanding Object-Oriented Development Practices

John W. Satzinger
Southwest Missouri State University

E. Reed Doke
Southwest Missouri State University

Bill C. Hardgrave
University of Arkansas

Toward a Framework for Understanding Object-Oriented Development Practices

John W. Satzinger
E. Reed Doke
Computer Information Systems Department
Southwest Missouri State University
Bill C. Hardgrave
Computer Information Systems and Quantitative Analysis
University of Arkansas

Abstract

This paper describes specific object-oriented system development cases that resulted from interviews with system developers in a variety of large Midwestern firms. Four object-oriented projects are described, reflecting the diversity of the types of systems and types of technology involved. A few typical issues are discussed. This is the first phase of a research project that seeks to understand issues related to object-oriented development. Additional research is planned to define and measure key development issues, and this will require developing a instrument based on the findings from these cases.

Introduction

An important issue facing most information systems developers is deciding how to approach the development of object-oriented systems. To date, information systems researchers have not provided much in the way of guidance, although initial research has focused on a variety of issues (Burkle et al., 1995; Curtis, 1995; Fayad et al., 1995; Fayad et al., 1996; Pancake, 1995). Object-oriented systems development methodologists, on the other hand, have provided a substantial amount of advice, primarily in books that propose complete OO development techniques and methods (e.g., Booch, 1994; Coad, 1997). Unfortunately, the methodologists do not usually address technology-specific issues, and they have not presented much evidence to demonstrate how well the techniques and methods actually work in practice.

This research is designed to take a broad look at OO system development practices to shed some light on how information systems developers are actually approaching the problem. Using this information, a stream of research can be developed that addresses both theoretical and practical concerns. Our initial focus is on the type of applications developed, the type of technology being used, and the staff involved in OO development. For example, in terms of applications, what are the types of applications developers have chosen for their pilot projects? What types of applications are currently underway? What applications are planned for the near future? In terms of technology, what methods, CASE tools, languages, and platforms are being used? In terms of staff, who are the people involved? What type of background do they have? How did they learn object-oriented development? Finally, we are interested in documenting some conclusions OO system developers may have reached based on their experiences. What seems to work? What does not? Further, what are the factors that have contributed to successful projects? What are some pitfalls to watch out for?

To answer these questions, we designed a multi-phase study to collect descriptive data from companies that are using object-oriented development. The first phase, reported in this paper, used phone interviews with open-ended questions to collect information about current object oriented development projects. Open ended questions under broad categories were used because so little previous research about object-oriented development was available to guide the development of an instrument. Additionally, because of the variety of expected approaches to object-oriented development, an open ended approach allowed the interview to focus on the issues of greatest importance in each case. The second phase will use these results to develop and use an instrument that will measure key issues about the practice of OO system development.

Results

Four case studies are briefly reported here. These cases were selected because they illustrate the diversity of applications and approaches in object-oriented development. Because of the diversity, this short paper mentions but cannot always define specific methods and technologies. Most are assumed to be widely known by interested readers.

Financial Services/Brokerage Firm.

This 15,000 employee, 650 office brokerage firm is developing a sales contact system as its first OO project. It is an extensive rewrite of a legacy system with approximately 200 users. The project has high visibility both within and outside of the IS organization, and it is considered important but not critical. The time to develop the system was planned at 1.5 to 2 years. The system resulted in 74 problem domain classes, which is the measure of size or scope used by the developers.

The analysis/design method used was an in-house variation of CRC combined with Coad. Technology used is Windows-NT with Visual Basic as the front-end to a relational database. Therefore, although the analysis and design used object-oriented methods, the implementation is only partially object-oriented. A CASE tool was not used.

The project had a staff of four - two systems analysts and two programmers. All four had traditional information systems backgrounds. They received in-house training, and outside consultants were not used. End users were involved only to approve the design, not as active participants. The project appears to be a success, although installation is not complete. The project staff found it easier to develop than they had expected, although they felt learning the object-oriented approach was the biggest hurdle.

Established National Telecommunications Firm.

This large telecommunications company has quite a few object-oriented applications completed, and two current projects were described. The focus here is a large telephone network management application that is highly visible and very critical. Most software development is for Windows NT clients. The development method includes both OMT and Jacobson use cases. They expect to move toward the unified Booch/OMT method in the near future. CASE tools have not been very useful, but they are hopeful about tools that support the unified method. C++, Smalltalk, and some Java are used for development. Legacy relational databases are used, but new components are object-oriented databases (OODBMS).

Staffing each project includes multiple "partitions" of 4 to 7 staff members that work on various phases of development. Training occurs by grouping staff with mentors. In-house training focuses on specific skills and technology. They were particularly pleased with the way Jacobson's use cases worked with users. The biggest problem seems to be lack of viable CASE tools.

Emerging National Telecommunications Firm.

This telecommunications firm has six object-oriented applications under development, which are similar in terms of technology and staffing. Most client workstations run Windows 95, and development is done under Windows NT. They use the Booch method for analysis and design, but also develop use cases early on with a problem domain expert. Iteration is crucial. C++ is used on all clients and servers (primarily Microsoft Visual C++). Internally developed "frameworks" (like coding templates) are used extensively, so programmers are able to focus on the application specific details. Databases are mostly relational.

They have had best results training staff with some but not too much experience programming in C. Some are hired in, and others are trained in house. Of the three months of formal training for all new hires, one month is devoted to OO and C++ specifically. Outside consultants have also been used, with mixed results. A great benefit of OO has been the improvement in communication with users because the focus stays on the user's vocabulary throughout. Problems occurred when consultants and "gurus" who did not actually

have the required skills or did not fit in the firm culture. The biggest problem mentioned was unrealistic user expectations.

National Insurance Firm.

This very large insurance company has an IS staff of 3000. IS managers recently pushed for and received an increased commitment to OO, which they feel is definitely the direction the company is going. One team is working on a comprehensive OO plan for the company overall. Another team is working to develop "frameworks" of company specific and industry specific classes to facilitate development and reuse. One IS division of 300 people has about 40 with training in OO, and entry level training now includes some OO for everyone, which will continue to increase. Several applications were discussed, and an electronic claim file application is included here. This involves enhancing a legacy application to add imaging capability. This is a very large, very visible, and very critical application now in pilot testing. Users are in 28 regional claims offices. Development time has been one year, with 8 months analysis and 4 months design and development. The pace has been rushed.

The project requires integrating tailorable software tools, including software that handles the imaging and software that handles work flow. SmallTalk and Visual Basic have also been used to customize and integrate. An informal development method based on Coad and also CRC is used, without a CASE tool. The project staff includes 40. Ten were involved in the OO aspects, and the others are installing hardware and completing other tasks. All OO staff were hand selected and trained, based on interests and skills, about half programmers and half previous support people. Many staff want to move into OO. There are no turnover problems, despite high stress levels because of the pace. Some problems include lack of specific development method. The biggest problem in this project stems from the need to integrate packages that take different approaches to OO, coupled with enhancing a legacy application. The typical methods and CASE tools do not help much with these problems, which are probably common to many IS projects.

Conclusions

The case studies completed to date confirm our assumption that object-oriented development needs to be studied using a broad perspective because of the great differences in the types of applications, the background of the people involved in development, and the type of technology being used. We are therefore working to develop an instrument to use to collect data that reflects the diversity of projects underway and completed.

The cases described here do illustrate some typical problems, though. For example, developers are not getting much support from CASE tools. But they are beginning to look to frameworks and class libraries for assistance. Additionally, they seem to be using a mixture of development methods and techniques, picked up from books and short training seminars. In many cases, the use case approach used in combination with object modeling seems to work. Similarly, projects typically involve several programming languages and/or packages that need to be integrated. Finally, it is clear that many projects will involve integration with existing legacy applications, and design methods that can help with integration are definitely needed.

References

Booch, G. (1994). *Object-Oriented Analysis and Design with Applications 2nd Ed*, Redwood City: Benjamin/Cummings

Burkle,U., Gryczan, G. & Zullighoven, H. (1995). Object-oriented system development in a banking project: methodology, experience, and conclusions. *Human-Computer Interaction*, 10, 293-336.

Coad, P. (1997). *Object Models: Strategies, Patterns, & Applications 2nd Ed*. Yourdon Press/ Prentice Hall.

Curtis, B. (1995). Objects of our desire: Empirical research on object-oriented development. *Human-Computer Interaction*, 10, 337-344.

Fayad, M. & Tsai, W. (1995). Object-oriented experiences. *Communications of the ACM*, 38 (10), 51-53.

Fayad, M., Tsai, W. & Fulghum, M. (1996). Transition to object-oriented software development. *Communications of the ACM*, 39 (2), 108-121.

Pancake, C. (1995). The promise and cost of object technology: A five year forecast. *Communications of the ACM*, 38 (10), 33-49.